

REMARKS

Claims 1-34 are pending in the current application. Applicants have amended claim 2 to correct a minor typographical error, and not for purposes relating to patentability. Reexamination and reconsideration of all of the claims are respectfully requested.

35 U.S.C. § 102

The Office Action rejected claims 29 and 30 under 35 U.S.C. §102 based on Kevner, U.S. Patent 5,956,509 (“Kevner”). Applicants respectfully traverse this rejection.

Kevner discloses a system and method for performing remote requests with an on-line service network. The Kevner remote request system and method is said to monitor and control the execution of remote requests on an on-line services network. (Abstract) When a remotely located client sends a remote request to the on-line service network, the remote request system monitors the remote request while returning operating control back to the client while the remote request remains pending in the on-line service network. (*Id.*) The remote request system also provides for the concurrent execution of multiple pending remote requests, provides status information about each remote request, provides for the cancellation of a pending remote request and optimizes the use of memory. (*Id.*) In addition, the remote request system dynamically allocates memory when data blocks of unknown size are transmitted over the on-line services network. (*Id.*)

What is missing from the Kevner reference includes, but is not necessarily limited to: (1) compiling a shell for communicating between a remote procedure and an agent API from remote procedure information; (2) compiling an API procedure from the remote procedure definition information; and (3) compiling an agent [remote] API from the remote procedure, API procedure, and shell. Simply put, while the Kevner design

references remote system calls and remote requests, the activity and procedures occurring at the remote site and the functionality at the remote site are relatively limited and fundamentally different from Applicants' design as claimed in claim 29.

Claim 29 requires, *inter alia*:

compiling a shell for communicating between the remote procedure (214) and an agent API (212) from the remote procedure information;

compiling an API procedure from the remote procedure definition information;
and

compiling an agent API (212) from the remote procedure (214), the API procedure, and the shell.

None of these limitations are present in Kevner. Kevner shows, in FIG. 2, a client 102 and server 104 arrangement connected by a gateway 124 and WAN 106, where different server applications 200b are executed. Kevner spells out the problem at Col. 2, ll. 34-46:

In many applications, users may request large amounts of data such as audio, multimedia, and large data files. While a remote procedure call allows the transfer of large blocks of data, *a conventional remote procedure call fails to provide timely status information about such data transfers. Consequently, the client cannot query the server regarding the status of the requested information.* Thus, a system may appear to "hang" when executing a remote procedure call over a wide area network connected by modems to telecommunication lines. Because users typically expect a high degree of user interaction, lack of status information over slower wide area networks means that many users will simply avoid requesting files with large amounts of data.

(emphasis added).

Kevner thus addresses the problem of the remote system failing to provide timely status information about data transfers. The present design

provides a system and method for executing a remote procedure in a remote processor from a client process on a local processor, not obtaining timely status information regarding data transfers. The present design is more involved and fundamentally different from the Kevner design.

Kevner goes on to discuss the relatively static and limited interaction between the Kevner client and the Kevner server with reference to FIG. 2. In essence, the client and server maintain static, pre-programmed routines, run time libraries, and other non-transferred and non-interactive functions, and do not compile a shell for communicating, API procedures, or an agent API:

Referring to FIG. 2, a block diagram of two of client applications 200a communicating with the service applications 200b is shown. The client applications 200a execute on the client processor 102 and the service applications 200b execute on the servers 120. *The client applications 200a communicate with the service applications 200b normally in the form of remote requests (i.e., a request which directs a service application 200b running on the server 120 to perform particular actions).*

...The client MPC [Microsoft Network Procedure Call] layer 206a and the service MPC layers 206b and 206c contain a variety of routines which allow the client and service applications to communicate with each other.

The client MPC layer 206a application programming interface resides in a dynamic link library. *The dynamic link library contains the set of routines which each application program uses to request and carry out remote requests. For example, as discussed in more detail below, during development of a client application 200a, a programmer writes software that calls or invokes the routines in the client MPC layer 206a. During execution of the program, the application retrieves the desired routine from the client MPC layer 206a dynamic link library and executes the routine like any other software routine.* The dynamic link libraries reduce storage space requirements because the dynamic link libraries allow each of the client applications 200a to share the routines in the client MPC layer 206a. The service MPC layer 206b exists in each Gateway 124 and the service MPC layer 206c exists on the servers 120. In the preferred embodiment, the

service MPC layer 206b existing in each Gateway 124 and the service MPC layer 206c existing in the servers 120 are identical.

Underneath the MPC layers 206a, 206b and 206c is a transport layer. In the preferred embodiment, the transport layer is called the Microsoft Network Connection Protocol or "MCP layer" and for convenience will be referred to as the MCP layer throughout the application. The MCP layer 210 also includes a client API ("the client MCP 210a") which runs on the client processors 102 and a server API ("the Gateway MCP 210b and the server MPC 210c") which runs on the Gateways 124 and the servers 120. The MCP layer 210 manages client-Gateway-Server communications that support simultaneous service sessions, allowing a client user to access multiple services simultaneously.

The Gateway MCP layer 210b then multiplexes and packetizes the messages and sends the messages over the local area network 122 to the server MCP layer 210c in the servers 120. The server MCP layer 210c in the servers 120, reformats the messages into function calls and passes the function calls to the service MPC layer 206c. *As discussed in more detail below, the service MPC layer 206c then directs the service applications 200b such as the CHAT service 202b and the WEATHER service 204b to execute the function calls and send responses back to the client applications 200a.*

When sending a response back to the client applications 200a, the service MPC layer 206c formats responses from the CHAT service 202b and the WEATHER service 204b into messages. The service MPC layer 206c then passes the messages to the server MCP layer 210c in that server 120. The server MCP layer 210c multiplexes (and packetizes) the messages and sends the multiplexed data over the local area network 122 to the Gateway MCP layer 210b. The Gateway MCP layer 210b then sends the multiplexed data over the wide area network 106 to the client MCP layer 210a in the client processor 102.

Col. 8, ll. 13-Col. 9, l. 19 (emphasis added).

Again, this simply states that remote requests are made via the request layer 206, and an application employs a set of routines to "carry out remote requests." No compiling of a shell for communicating between the remote procedure and an agent API, no compiling API procedures from remote procedure definition information, and no

compiling of an agent API from the remote procedure, the API procedure, and the shell as required by the present claims 29 and 30.

Further sections of the Kevner patent similarly offer limited functionality:

Client applications 200a make use of a high-level client MPC layer 206a via an application programming interface (API) which is optimized to permit efficient client-server communications over relatively slow/high latency wide area networks 106. In the preferred embodiment, the client contains the client MPC layer 206a while the Gateways 124 and the service applications contain the service MPC layers 206c.

The MPC layers 206a, 206b and 206c are similar to the session layer defined by the International Standards Organization (ISO) Open Systems Interconnection (OSI) Reference Model. *The MPC layers 206a, 206b and 206c contain routines which allow the client applications 200a and service applications 200b to send and receive function calls.*

*In the preferred embodiment, when a programmer develops the client application 200a, the programmer adds software that communicates with the client MPC layer 206a. Furthermore, when a programmer develops the service application 200b, the programmer adds software that communicates with the service MPC layer 206c. For example, to develop the client application 200a, the programmer inserts code that "calls" or invokes the routines in the client MPC layer 206a. **The routines in the client MPC layer 206a then execute the necessary instructions to create a remote request and to send the remote request to the other layers in the on-line services network 100.** Consequently, the programmer does not need to know the details about the operation of other layers in the on-line services network 100.*

In the preferred embodiment, the MPC layers 206a, 206b and 206c do much more than send remote messages to each other. The MPC layers 206a and 206c create internal data structures for monitoring pending remote requests. In addition, the MPC layers 206a and 206c create remote request identifiers which uniquely identify each remote request. The MPC layers 206a and 206c use the identifiers to properly route the remote requests to their proper destinations. As described in more detail below, the MPC layers 206a, 206b and 206c, in the present invention, are compatible with

the Object Linking and Embedding (OLE) 2.0 architecture defined by Microsoft Corporation. The Object Linking and Embedding (OLE) 2.0 architecture routine is well known in the art, and is described in OLE 2 Programmer's Reference Vol. I, Microsoft Press, 1993, and in OLE 2 Programmer's Reference Vol. II, Microsoft Press, 1993. Also, while the following description describes the MPC layers 206a, 206b and 206c in object-oriented terminology, a person of ordinary skill in the art will appreciate that other programming techniques can be used to implement the MPC layers 206a, 206b and 206c without using an object-oriented programming language.

Col. 10, ll. 14-63 (emphasis added).

Again, this discusses the static interaction between client and server, where issuing a function call at the client results in the server executing a function programmed thereon. Kevner does state that the MPC layers “create internal data structures for monitoring pending remote requests.” However, such internal data structures differ from comiling shells, compiling an API “procedure,” and compiling an agent from the remote procedure, the API procedure, and the shell.

The Office Action attempts to correlate the claim limitations with Kevner as follows:

“compiling a shell for communicating between the remote procedure (214) and an agent API (212) from the remote procedure information (Abstract, col. 1, lines 33-50)”
Office Action, p. 3

As noted above, the Abstract broadly states that the remote request system “monitors and controls the execution of remote requests on an on-line services network.” Compiling a shell and an agent API from remote procedure information is neither suggested nor disclosed. Col. 1, ll. 33-50 generally describes remote procedure calls and the standards governing remote procedure calls, stating in part “an application programmer does not need to write software to transmit computational or Input/Output-related requests across a network, to handle protocols and to deal with network errors.”

Again, compiling a shell and an agent API from remote procedure information is neither suggested nor disclosed by the cited text.

“compiling an API procedure from the remote procedure identification information (col. 2, lines 35-45)” Office Action, p. 3.

Col. 2, ll. 35-45 of Kevner, reprinted above, simply outlines the purported shortcomings of prior data transfer designs, particularly highlighting the “[failure] to provide timely status information about such data transfers,” and the “lack of status information.” These passages do not in any way show compiling an API procedure from remote procedure information. The cited passage deals with providing status information, previous systems not providing this status information, and not compiling an API procedure, particularly from remote procedure identification information.

“compiling an agent API (212) from the remote procedure (214), the API procedure, and the shell ([A]bstract, col. 1, ll. 33-50).” Office Action, p. 3.

The Kevner Abstract states that “the remotely located client sends a remote request to the on-line service network, the remote request system monitors the remote request while returning operating control back to the client while the remote request remains pending in the on-line service network.” Thus a request is sent to the remote system, control passes to the client, and the remote system operates on the request. This materially differs from compiling an agent API from the remote procedure, API procedure, and the shell. Col. 1, ll. 33-50 discusses operation of remote procedure calls, which materially differ from compiling an agent API from the remote procedure, API procedure, and shell.

In short, none of these passages demonstrate the novel aspects presented in the claims. Claim 29 is simply not anticipated by the Kevner reference, as Kevner neither suggests nor discloses the unique aspects of claim 29 identified above. Claim 30 is not anticipated by Kevner as it depends from allowable claim 29.

35 U.S.C. § 103

The Office Action rejects claims 1-28 and 31-34, which includes independent claims 1, 8, 13, 18, 25, 28, 31, and 34 under 35 U.S.C § 103 based on Kevner in view of Tran, U.S. Patent 6,202,060 (“Tran”). Applicants respectfully traverse this rejection.

Generally, independent claims 1, 8, 13 18, 25, 28, 31, and 34 include verbiage materially different from the disclosures of Kevner and/or Tran, either alone or in combination. Furthermore, there is simply no motivation to combine Kevner with Tran in the manner suggested without resorting to the use of hindsight, which is improper.

Claim 1

Claim 1 requires, *inter alia*, translating a remote procedure call into a CGI compatible information transfer protocol. The Office Action recognizes that Kevner does not mention the concept of CGI, specifically claimed in this and all independent claims discussed below.

The Office Action states that Tran teaches “a CGI-compatible transfer protocol,” citing col. 26, ll. 25-30 and col. 28, ll. 30-40. While these passages in Tran describe the Common Gateway Interface (CGI), use of CGI in Tran is in a fundamentally different manner than the claimed “translating a remote call procedure into a CGI compatible information transfer protocol.” Tran, in the passages cited, passes a query to a server in a query string containing the name of a CGI script. Tran, ll. 24-27. The CGI script sends the search to a database located on the server, receives the result of the query (in addition to the HTML page created by the database), and passes the query back to the server. *Id.* at ll. 27-30. Tran provides a data filter, stating the data filter reduces the amount of documents to be viewed. *Id.* at col. 28, ll. 35-37. “The browser can be configured to download specific objects, such as text...” *Id.* at ll. 37-40.

Applicants contend that the most that can be said of the cited passages in Tran is that certain databases are compatible with CGI. Applicants contend, first, that the

limitations presented in claim 1 are not disclosed nor suggested by Tran, and second, that combining the CGI aspects of Tran with the Kevner design is improper.

The actual claim limitation at issue is “translating a remote procedure call into a CGI compatible information transfer protocol.” Kevner cannot translate a remote procedure call into a CGI compatible information transfer protocol because it admittedly does not employ CGI in any manner. Tran cannot be said to translate a remote procedure call into a CGI compatible information transfer protocol, because Tran passes a query to a server in a query string including the name of a CGI script, and the CGI script sends the search to a database, receives the result of the query, and passes the result back to the server. No translation occurs, either expressly or impliedly, in Tran. Thus Tran does not include the limitation.

As neither reference includes the limitation of “translating a remote procedure call into a CGI compatible information transfer protocol,” one element of the claim is missing from the cited references. Absence of an element from the cited references indicates claim 1 is nonobvious based on these references.

Further, there is no motivation to combine the remote request system of Kevner with the CGI aspects recited in Tran. The motivation to combine is alleged to be “because it would have an efficient system that could provide specific functions that transferring information between a World Wide Web server and any program is designed to accept and return data that conforms to the CGI specification.” (Office Action, p. 5; see also, pp. 7, 8, 9, 10, 12, 13, 14, 15, 16). This alleged motivation reads more into either reference than is present in the references themselves, and merely states a broad end result rather than a motivation to combine the teachings of the Tran and Kevner references. In reality, there is no suggestion in Kevner to use CGI, nor any suggestion in Tran to employ remote requests in a network in the manner presented in Kevner.

Applicants thus further dispute the combination of Kevner and Tran in the manner suggested in the Office Action. Applicants respectfully submit that no motivation to

combine the references in the manner suggested is presented within the references themselves. The statement in the Office Action regarding the motivation to combine being for the purpose of “provid[ing] specific functions that transfer[] information” does not demonstrate any motivation to combine the references explicit or implied within the two references themselves, but instead takes the CGI aspect of Tran and alleges that it could be applied somehow to Kevner. Applicants contend that simply no motivation exists to employ the CGI aspect of Tran in Kevner, and certainly no motivation to employ the remote request system of Kevner is included in Tran.

The Federal Circuit has held that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention, absent some teaching, suggestion or incentive supporting the combination. *ACS Hospital System, Inc. v. Montefiore Hospital*, 732 F.2d 1572 (Fed. Cir. 1984). Without some showing in the prior art that suggests in some way a combination in order to arrive at the claimed invention, it is impermissible to use the Applicants’ teaching to search references for the claimed elements and combine them as claimed. *In Re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991); *In Re Laskowski*, 871 F.2d 115, 117 (Fed. Cir. 1989); *see also, Ex Parte Lange*, 72 U.S.P.Q. 90, 91 (C.C.P.A. 1947) (“It seems to us that the Examiner is using appellant’s disclosure for the suggestion of the combination since there is no suggestion in any of the patents for their combination in the manner claimed by Applicant.”); *In re Leonor*, 158 U.S.P.Q. 20, 21 (C.C.P.A. 1968) (the issue is “whether teachings of prior art would, of themselves, and without benefit of applicant’s disclosure, suggest [a process] which would make claimed invention obvious...” (emphasis in original)). As noted, the Tran reference does not suggest combining the CGI script design disclosed with the remote request aspect of Kevner to produce the unique method claimed in Applicants’ independent claim 1.

Applicants respectfully submit that the Office Action uses hindsight in rejecting the claims herein. It is only through hindsight, after seeing Applicants’ disclosure, that it would be considered possible to create the access control method as claimed by the Applicants. With regard to the use of hindsight, or the use of an Applicant’s teaching to combine

references, the courts have overwhelmingly condemned such combinations and have upheld the validity of patents or claims of patents in which such hindsight was employed to combine the references. *W.L. Gore Associates, Inc. v. Garlock, Inc.*, 220 U.S.P.Q. 303, 313 (Fed. Cir. 1983), (condemning the “insidious effect of a hindsight syndrome wherein that which only the inventor taught is used against its teacher”); *In re Fine*, 837 F.2d 1044, 1051 (Fed. Cir. 1988) (“One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention.”) Applicants respectfully submit that combination of aspects of the Tran reference with the Kevner design is merely a hindsight reconstruction of the invention using Applicants’ disclosure and attempting to use Applicants’ claims as a guide. Such hindsight reconstruction of the claimed system is inappropriate and thus rejection of the independent claim 1 for this reason is improper.

Based upon the totality of the foregoing, Applicants respectfully submit that claim 1 is allowable over the references of record, and that all claims dependent from claim 1 are allowable as they depend from an allowable base claim.

Remaining Independent Claims

The remaining independent claims 8, 13, 18, 25, 28, 31, and 34 contain CGI limitations similar to those found in claim 1. As a result, the combination of Tran and Kevner do not disclose the limitations claimed, and further, no motivation exists to combine Tran and Kevner in the manner suggested.

Claim 8

With respect to claim 8, the limitation “translating the remote procedure call into a CGI compatible information transfer protocol” is neither disclosed nor suggested by Kevner and/or Tran, either alone or in combination. Further, as with claim 1, no motivation exists to combine the cited references to produce the claimed invention, and it is only with the use of hindsight that claim 8 can be rejected in the manner presented,

which is improper. Claim 8 is allowable over the references of record, and that all claims dependent from claim 8 are allowable as they depend from an allowable base claim.

Claim 13

Claim 13 requires “accepting a remote procedure call, *the remote procedure call transmitted in a CGI-compatible information transfer protocol*, the remote procedure call having information identifying the remote procedure (214) including a global remote procedure locator.” (emphasis added). As with the preceding claims, this limitation is neither disclosed nor suggested by Kevner and/or Tran, either alone or in combination. No motivation exists to combine the cited references to produce the claimed invention, and it is only with the use of hindsight that claim 13 can be rejected in the manner presented, which is improper. Claim 13 is therefore allowable over the references of record, and that all claims dependent from claim 13 are allowable as they depend from an allowable base claim.

Claim 18

Claim 18 recites a gateway “*for translating a remote procedure call having information identifying the remote procedure (214) including a global remote procedure locator into a CGI-compatible information transfer protocol*, for transmitting the translated remote procedure call to the remote processor (220), and for interpreting a remote procedure response into a client process-compatible format” and “for interpreting a remote procedure call translated by the client application program interface (206) into a remote procedure-compatible format, for invoking the remote procedure (214) in the remote processor (220) to produce a remote procedure response, and *for translating a remote procedure response into the CGI-compatible information transfer protocol*.” (emphasis added). Neither Tran nor Kevner, alone or in combination, translates a remote procedure into a CGI compatible format, nor a remote procedure response into the CGI-compatible information transfer protocol as required by claim 18. No motivation exists to combine the cited references to produce the claimed invention, and it is only with the

use of hindsight that claim 18 can be rejected in the manner presented, which is improper. Claim 18 is therefore allowable over the references of record, and that all claims dependent from claim 18 are allowable as they depend from an allowable base claim.

Claim 25

Claim 25 claims “a client application program interface (206) for translating a remote procedure call having information identifying the remote procedure (214) including a global remote procedure locator into a CGI-compatible information transfer protocol” and “for translating the remote procedure response into the CGI-compatible information transfer protocol.” As with claim 18, neither translating a remote procedure call into a CGI-compatible information transfer protocol nor translating the remote procedure response into the CGI-compatible information transfer protocol” is suggested or disclosed by Kevner and/or Tran, either alone or in combination.

No motivation exists to combine the cited references to produce the claimed invention, and it is only with the use of hindsight that claim 25 can be rejected in the manner presented, which is improper. Claim 25 is therefore allowable over the references of record, and that all claims dependent from claim 25 are allowable as they depend from an allowable base claim.

Claim 28

Claim 28 recites “an agent application program interface (212) for ... translating a remote procedure response into a CGI-compatible information transfer protocol; ... wherein the agent application program interface (212) is communicatively coupled to a client application program interface (206) for translating a remote procedure call having information identifying the remote procedure (214) including a global remote procedure locator into a CGI-compatible information transfer protocol for transmitting the translated remote procedure call to the remote processor (220); ...” Neither the Kevner reference nor the Tran reference relied on in the Office Action, either alone or in combination,

disclose or suggest translating a remote procedure response into a CGI-compatible information transfer protocol, or translating a remote procedure call having information identifying the remote procedure including a global remote procedure locator into a CGI-compatible information transfer protocol. No motivation exists to combine the cited references to produce the claimed invention, and it is only with the use of hindsight that claim 28 can be rejected in the manner presented, which is improper. Claim 28 is therefore allowable over the references of record.

Claim 31

Claim 31 includes the following limitation: “means for translating the remote procedure call into a CGI-compatible information transfer protocol.” As noted above, neither Kevner nor Tran, either alone or in combination, disclose or suggest translating a remote procedure call into a CGI-compatible information transfer protocol, and neither disclose nor suggest the means for doing so. No motivation exists to combine the cited references to produce the claimed invention, and it is only with the use of hindsight that claim 31 can be rejected in the manner presented, which is improper. Claim 31 is therefore allowable over the references of record.

Claim 34

Claim 31 requires “translating the remote procedure call into a CGI-compatible information transfer protocol.” As noted above, neither Kevner nor Tran, either alone or in combination, disclose or suggest translating a remote procedure call into a CGI-compatible information transfer protocol, and neither disclose nor suggest the means for doing so. No motivation exists to combine the cited references to produce the claimed invention, and it is only with the use of hindsight that claim 34 can be rejected in the manner presented, which is improper. Claim 34 is therefore allowable over the references of record.

Accordingly, it is respectfully submitted that all pending claims fully comply with 35 U.S.C. § 103.

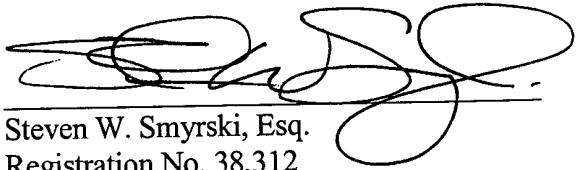
CONCLUSION

In view of the foregoing, it is respectfully submitted that all claims of the present application are in condition for allowance. Consideration of all of the claims, as amended, is respectfully requested and allowance of all pending claims at an early date is solicited.

Applicants believe that no fees are due in accordance with this Amendment beyond those included herewith. Should any additional fees be due, the Commissioner is hereby authorized to charge any deficiencies or credit any overpayment to Deposit Account 502026.

Respectfully submitted,

Date: October 29, 2004


Steven W. Smyrski, Esq.
Registration No. 38,312

SMYRSKI & LIVESAY, LLP
3310 Airport Avenue, SW
Santa Monica, California 90405-6118
Phone: 310.397.9118
Fax: 310.397.9158